(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－307817

(43)公開日　平成10年(1998)11月17日

(51)Int.Cl.[6]	識別記号	ＦＩ		
Ｇ０６Ｆ　17/21		Ｇ０６Ｆ　15/20	５６６Ｍ	
Ｂ４１Ｊ　21/00		Ｂ４１Ｊ　21/00	Ａ	
Ｇ０６Ｆ　3/12		Ｇ０６Ｆ　3/12	Ｆ	
		15/20	５４８Ｅ	

審査請求　未請求　請求項の数4　ＯＬ　（全 13 頁）

(21)出願番号　特願平9－118412

(22)出願日　平成９年(1997)５月８日

(71)出願人　000104174
カシオ電子工業株式会社
埼玉県人間市宮寺4084番地

(71)出願人　000001443
カシオ計算機株式会社
東京都渋谷区本町１丁目６番２号

(72)発明者　小林　正樹
東京都東大和市桜が丘２丁目229　番地
カシオ電子工業株式会社内

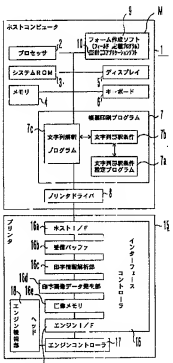
(74)代理人　弁理士　大菅　義之

(54)【発明の名称】　文字列変換装置

(57)【要約】

【課題】　本発明は表計算やデータベースソフトを使用し、またフォームオーバレイ印字を行う際使用する文字列の解釈条件の設定が可能な文字列変換装置に関し、特にデータに基づく色分け処理等の印字加工処理を正確に行うことができる文字列変換装置を提供するものである。

【解決手段】　本発明は、例えばフォームオーバレイ処理により帳票を作成する際、フォーム作成ソフトにより帳票のフォームデータを作成し、例えば表計算のアプリケーションで作成したＣＳＶデータを被フォームデータとして帳票印刷プログラム７により合成する。この際、ユーザの登録した文字の解釈条件に従ってＣＳＶデータの文字列を解釈し、フォームオーバレイ印刷を行うものであり、ユーザの希望する解釈条件を特定の文字に対し設定でき、この設定に従って処理するので例えば数値や文字列に従った色分け処理等を容易に行うことができる。



【特許請求の範囲】

【請求項１】　入力する文字列を解釈する条件を予め設定する文字列解釈条件設定手段と、

該文字列解釈条件設定手段によって設定された解釈条件に従って入力する文字列データを解釈する文字列解釈手段と、

特定の印字領域に印字されるべく入力した文字列を予め決められた所定の条件に従って識別する識別手段を有し、該識別手段の識別結果に応じて特定の印字加工処理を行う加工処理手段と、

前記識別手段は前記文字列解釈手段により解釈された結果に基づいて前記入力文字列を識別することを特徴とする文字列変換装置。

【請求項２】前記所定の条件は、フォームオーバレイ印刷の帳票定義データに定義されることを特徴とする請求項１記載の文字列変換装置。

【請求項３】前記文字列は表計算アプリケーションソフトにより生成されたＣＳＶデータであることを特徴とする請求項１記載の文字列変換装置。

【請求項４】前記加工処理手段で加工処理されたデータは、記録装置又は表示装置に出力されることを特徴とする文字列変換装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は表計算やデータベースソフトを使用し、またフォームオーバレイ印字を行う際使用する文字の解釈条件の設定が可能な文字列変換装置に関する。

【０００２】

【従来の技術】従来、表計算ソフトやデータベースソフトを使用してコンピュータを駆動する際、各種記号を含む文字列又は数値列を解釈して処理を行っている。また、例えばコンピュータにプリンタ装置を接続しフォームオーバレイ印字を行う際、各種記号を含む文字列を解釈して印字処理を行っている。

【０００３】かかる場合、上述のような文字列又は数値列データ（以下単に文字列データという）の解釈処理は、予め持ち合わせているプログラムに従って文字列データが本来持つ意味として解釈され、処理されている。したがって、文字列データの解釈処理は、プログラムが予め持ち合わせている、文字列を解釈する際に必要な条件の情報に従うしかなかった。

【０００４】例えば、文字列“◆100,000 ”を解釈する場合、通常「◆」は負符号として解釈され、また桁区切り記号「，」は無視される。したがって、この場合、上述の文字列“◆100,000 ”は「－100000」と解釈される。また、文字列“□東□京”では、「□」は全角スペースを表し、通常全角スペースは無視されるため、「□」の記号を除いた「東京」と解釈される。尚、図１１は通常行われる「文字列解釈条件」の一例を示す。

【０００５】

【発明が解決しようとする課題】従来の記録システムでは上述のように、文字列データの解釈処理を行う場合、使用するプログラムが予め持ち合わせている「文字列解釈条件」に従った解釈処理しかできなかった。このため、当該プログラムが通常持ち合わせている「文字列解釈条件」に適合しない文字列を解釈する場合、期待通りの結果が得られなかった。

【０００６】例えば、ある文字列データにおいて「●」の記号を負符号と解釈したい場合、つまり文字列“●100,000 ”を「－100000」と解釈したい場合、従来例では「●」の記号が「文字列解釈条件」に定義されていないため、“●100,000 ”を「－100000」と解釈することはできなかった。図１２は、前述の３例を模式的に示す図であり、“◆100,000 ”や“□東□京”の文字列に対しては正確な解釈処理が可能であるが、“●100,000 ”の文字列に対しては正確な解釈ができない。

【０００７】一方、例えば桁区切り記号「，」はＣＳＶ形式のフォーマットではレコード単位の区切りとして認識される。したがって、このようなフォーマットで長い桁の数値を表現する場合問題となる。特に、ＣＳＶデータを被フォームデータとして帳票フォームに印字する場合、被フォームデータ（ＣＳＶデータ）の数値によって各項目（各セル）に対する色分処理やその他の印字加工処理を行うことがあり、正確に被フォームデータを読みとることが重要である。

【０００８】例えば、帳票フォームの各セルにマイナス（－）100,000 以下のデータを印字する場合、赤色でセルの背景を塗り分ける加工条件が設定されているとすれば、正確にマイナス（－）を認識する必要がある。かかる場合、上述のように“◆100,000 ”や“●100,000 ”を「－100000」と解釈できなければ正確な色分け処理を行うことはできない。

【０００９】また、数値に限らずＣＳＶデータに含まれる地名に対する色分け加工条件の場合でも、正しく「東京」や「札幌」、「大阪」と認識できなければ正確な色分け加工を行うことができない。

【００１０】尚、上述の例は帳票フォーム、及びＣＳＶデータについて特に説明したが、他の形式のフォーム、及び他の形式のデータであっても同様である。本発明は上記課題を解決するため、「文字列解釈条件」をユーザそれぞれの必要により任意に設定可能とし、文字列データを解析する際、ユーザの期待通りの結果が得られ、且つデータに基づく色分け処理等の印字加工処理を正確に行うことができる文字列変換装置を提供するものである。

【００１１】

【課題を解決するための手段】上記課題を解決するため、請求項１記載の発明によれば、入力する文字列を解釈する条件を予め設定する文字列解釈条件設定手段と、

該文字列解釈条件設定手段によって設定された解釈条件に従って入力する文字列データを解釈する文字列解釈手段と、特定の印字領域に印字されるべく入力した文字列を予め決められた所定の条件に従って識別する識別手段を有し、該識別手段の識別結果に応じて特定の印字加工処理を行う加工処理手段と、前記識別手段は前記文字列解釈手段により解釈された結果に基づいて前記入力文字列を識別する文字列変換装置を提供することによって達成できる。

【００１２】ここで、文字列解釈条件設定手段は、例えば所定のプログラムに従って特定の文字に対し解釈条件の設定処理を行う。また、ここで設定される文字は、例えば「□」や「◆」、「，」等の文字と異なり、具体的にユーザの必要に応じて設定するものである。例えば、文字列解釈時の削除文字として「★」を設定し、数値解釈時の負符号文字として「●」を設定し、同じく数値解釈時の小数点文字として「｜」を設定する。

【００１３】このように構成することにより、通常設定されている文字の解釈条件ではない解釈条件に従って文字列を解釈させる必要が生じた場合、すなわちユーザの必要に応じて、通常の文字列解釈条件と異なる条件で特定の文字列の解釈を行うことができる。

【００１４】一方、識別手段は前記文字列解釈手段により解釈された結果に基づいて前記入力文字列を識別するものであり、例えば上記例のように「●」を数値解釈時の負符号文字と解釈する設定を行った場合、「●50」は「－50」と解釈する。したがって、数値や文字列を正確に判断することができ、上記加工処理の際、すなわち数値の大小や文字列の内容によって色分け印字処理等を行う際、数値や文字列を正確に判断することができ、正確な色分け処理等を行うことができる。

【００１５】請求項２の記載は、上記請求項１記載の発明をより具体的に示すものであり、前記所定の条件は、フォームオーバレイ印刷の帳票定義データに定義される構成である。

【００１６】すなわち、フォームオーバレイ印刷の際、被フォームデータとして供給されるデータに上記文字列や数値が含まれている場合、ユーザの設定した文字に対し、対応する解釈を行って印字処理を行うので、例えば数値の大小等による色分け処理等の印字条件に従ったフォームオーバレイ印刷を行うことができる。

【００１７】請求項３の記載は、上記請求項１記載の発明をより具体的に示すものであり、前記文字列は表計算アプリケーションソフトにより生成されたＣＳＶデータである。

【００１８】すなわち、ＣＳＶデータは「，」を桁区切り記号と判断するが、例えばユーザが「｜」を数値解釈時の小数点文字として予め登録しておくことにより、数値の桁数を正確に認識することができ、例えば上記のように数値の大小等による色分け処理を行う場合には、ＣＳＶデータに対しても正確な色分け処理を行うことができる。

【００１９】請求項４の記載は、上記請求項１記載の発明をより具体的に示すものであり、前記加工処理の出力は記録装置又は表示装置に出力される構成である。したがって、例えばプリンタ装置等の記録装置に出力する場合、記録装置内の例えば印字情報解析部で解析処理し、画像メモリに一旦展開した後記録紙に印字出力する。また、ＣＲＴディスプレイ等の表示装置に出力する場合、表示装置内の例えばＶＲＡＭに表示データとして出力し、表示処理を行う。

【００２０】ここで、例えば表示装置に対し本例の文字列変換装置で変換処理したデータを表示する場合として、帳票フォーム等のフォームオーバレイ画像の表示以外に、データベース、表計算等の画像を表示することができる。

【００２１】

【発明の実施の形態】以下、本発明の実施形態例を図面を用いて詳細に説明する。図１は本例の文字列変換装置を説明するシステム構成図である。同図において、本システムはホストコンピュータ１とプリンタ装置１５で構成されている。ホストコンピュータ１はプロセッサ２、システムＲＯＭ３、メモリ４、ディスプレイ５、キーボード６、帳票印刷プログラム７、プリンタドライバ８で構成されている。また、ホストコンピュータ１は、ハードディスクＭを装備し、その内部にはフォーム作成ソフト（フィールド定義プログラム）９、表計算アプリケーションソフト１０等のユーティリティソフトやアプリケーションソフトが記憶されており、プロセッサ２はこれらのプログラムを実行することにより種々の情報を処理する事ができる。

【００２２】プロセッサ２はシステムＲＯＭ３に記憶されたシステムプログラムに従って処理を行い、例えばハードディスク等からフォーム作成ソフト９を読み出し帳票フォームの作成処理を行う。また、表計算アプリケーションソフト１０を読み出し、表計算処理を行う。

【００２３】また、プロセッサ２が行う上述の処理の間、発生するデータはメモリ４のワークエリアに格納され、演算処理等に使用される。一方、ディスプレイ５には必要なデータが表示され、キーボード６からキー操作信号がプロセッサ２に供給される。

【００２４】また、帳票印刷プログラム７は文字列解釈条件設定プログラム７ａ、文字列解釈条件７ｂ、文字列解釈プログラム７ｃで構成されている。尚、この文字列解釈条件設定プログラム７ａ、文字列解釈条件７ｂ、文字列解釈プログラム７ｃの具体的な処理については後述する。

【００２５】プリンタドライバ８はホストコンピュータ１に接続するプリンタ装置１５の機種に対応した入出力制御を行い、後述する被フォームデータをプリンタ装置

１５に出力する。

【００２６】一方、プリンタ装置１５はインターフェイスコントローラ１６、エンジンコントローラ１７、印字ヘッドを含むエンジン機構部１８で構成されている。インターフェイスコントローラ１６はホストインターフェイス（以下、ホストＩ／Ｆという）１６ａ、受信バッファ１６ｂ、印字情報解析部１６ｃ、印字画像データ発生部１６ｄ、画像メモリ１６ｅ、エンジンＩ／Ｆ１６ｆで構成されている。

【００２７】図２は上記システムの中の、特にホストコンピュータ１の機能構成を説明する図である。すなわち、図２に示すソフトやファイルは、図１に示したフォーム作成ソフト９、表計算アプリケーションソフト１０、帳票印刷プログラム７の機能的な関係を示す図である。尚、本例ではフォームオーバレイ印字の例を説明し、特に当該フォームオーバレイ印字で使用する被フォームデータとして、表計算アプリケーション１０に登録されたデータをＣＳＶ形式のデータ（以下、ＣＳＶデータという）として使用する。

【００２８】したがって、本例ではフォーム作成ソフト９は、帳票フォームを作成するソフトであり、図２に示すように具体的にはＦＬ４ファイル１１ａとＥＧ４ファイル１１ｂを作成する。ここで、ＦＬ４ファイル１１ａはフォームコマンドファイルであり、フォーム作成ソフト９で作成した帳票フォーム（帳票フォームの図形）をフォーム作成ソフト独自のコマンドに置き換えるファイルであり、例えばテキスト形式で記述される。具体的には図３の（ａ）に示すような帳票フォームを作成する際、罫線の線種や矩形の座標位置等のデータであり、例えば矩形の場合「ＢＯＸ4.0,2.0,56.0,26.0,...」と記述され、罫線の場合「ＬＩＮＥ4.0,4.0,56.0,4.0,...」と記述される。

【００２９】一方、ＥＧ４ファイル１１ｂはフォームデータファイルであり、フォーム作成ソフト９で作成した帳票フォームをプリンタ装置１５のコマンドに置き換えて記憶する。具体的にはバイナリ形式のデータで記述し、例えば矩形の場合「1Dh,00h,3Bh,00h,3Bh,40h,3Bh,...」と記述する。

【００３０】次に、フィールド定義プログラム１２は上述の帳票フォームに書き込む被フォームデータの書き込み項目を定義するプログラムである。具体的には、図３の（ｂ）に示すように各項目を定義する。例えば、帳票フォームのナンバー（Ｎｏ）の領域（フィールド）がフィールド１Ａである。また、帳票フォームの各月を表示する領域がフィールド２Ａ～２Ｃであり、以下各項目毎に、フィールド３Ａ～３Ｅ、フィールド４Ａ～４Ｅ、フィールド５Ａ～５Ｅ、フィールド６Ａ～６Ｅと続く。また、フィールドファイル（ＦＬＤファイル）１３には上述のフィールド定義プログラム１２によって作成されたフィールド定義情報ファイルが登録される。

【００３１】図４は、更に上述の表計算アプリケーションソフト８、及び帳票印刷プログラム７の詳しい機能構成を説明する図である。ここで、プログラムＡは上述の表計算アプリケーションソフト８の処理プログラムの一部を示し、レコード表示プログラムも表計算アプリケーションソフト８の一部の構成を示す。また、ファイルＡは表計算アプリケーションソフト８によって作成されるＣＳＶデータの記憶ファイルを示す。すなわち、ファイルＡには表計算アプリケーションソフト８によって作成するＣＳＶデータが格納され、例えばレコード単位に分割され、分割データの形態で格納されている。

【００３２】また、文字列解釈条件設定プログラム７ａは文字列の解釈条件を規定するプログラムであり、上述のレコード単位の文字列データを解釈する際に必要となる。また、文字列解釈条件設定プログラム７ａに基づいて設定される解釈条件は文字列解釈条件記憶エリア７ｂに登録される。

【００３３】図５は文字列解釈条件記憶エリア７ｂのメモリマップを説明する図である。同図に示すように、文字列解釈条件記憶エリア７ｂは「文字列解釈時削除文字数（ＱＣＳ）」、「数値解釈時削除文字数（ＱＮＳ）」、「数値解釈時負符号文字数（ＱＮＭ）」、「数値解釈時少数点文字数（ＱＮＰ）」の各データ記憶エリア２５と、これらの各記憶エリアに記憶されるデータの具体的な文字の記憶エリアで構成されている。例えば、記憶エリア２５ａは文字列解釈時の具体的な削除文字を＃１、＃２、・・・に記憶し、記憶エリア２５ｂは数値解釈時の具体的な削除文字を＃１、＃２、・・・に記憶し、記憶エリア２５ｃは数値解釈時の具体的な負符号文字を＃１、＃２、・・・に記憶し、記憶エリア２５ｄは数値解釈時の具体的な小数点文字を＃１、＃２、・・・に記憶する。

【００３４】一方、文字列解釈プログラム７ｃはファイルＡに記憶されるＣＳＶデータの文字列を上述の文字列解釈条件記憶エリア７ｂに記憶された解釈条件に従って解釈し、ファイルＢに解釈後のデータを書き込む。また、プログラムＢはファイルＢに書き込んだデータを図２に示すプリンタドライバ８に出力するプログラムである。

【００３５】以上の構成のシステムにおいて、先ず文字列解釈条件設定プログラム７ａに従った文字列解釈条件の設定処理を説明する。この処理は、前述のホストコンピュータ１に設けられたキーボード６を操作し、特別な解釈条件を必要とする文字の設定を行う。具体的には、文字列解釈条件設定プログラム７ａの処理により、ディスプレイ４上に例えば図６に示す表示を行う。すなわち、本例の場合、文字列解釈時削除文字の設定、数値解釈時削除文字の設定、数値解釈時負符号文字の設定、数値解釈時小数点文字の設定の４項目であり、それぞれの設定項目に対して設定処理を行う。

【００３６】例えば、文字列解釈時の削除文字として通常使用される「□」、「II」（但し、「II」は半角スペースを意味する。以下便宜上半角スペースをこのように表記する）以外に、他の文字を削除文字として登録する場合、図６に示す文字列削除文字（Ｓ）の欄に登録を希望する文字、例えば「★」、「☆」を設定する。また、数値列の削除文字（Ｎ）に対しても同様の処理を行い、ユーザが削除文字の登録を必要とする文字があれば設定する。但し、本例の説明ではユーザが希望する特別な数値列に対する削除文字の設定はないものとする。

【００３７】次に、数値負符号文字の設定処理を行う。ここで、例えば負符号文字として通常使用される「◆」等以外に、「●」を負符号文字として設定しようとする場合、図６に示す数値負符号文字（Ｍ）の欄に登録を希望する「●」の設定を行う。さらに、「｜」を数値小数点文字として設定する場合、同図に示す数値小数点文字（Ｐ）の欄に登録を希望する「｜」の設定処理を行う。

【００３８】尚、上述の設定処理の際、誤ったデータを入力した場合、同図に示す「取消」ボタン２６ｂを押してそれまでの入力操作を取り消して元に戻すことができる。以上のようにして解釈条件設定処理が終了すると、「了解」ボタン２６ａを押下して処理を終了する。したがって、上述の処理により文字列解釈条件記憶エリア７ｂには図７に示す解釈条件が設定される。すなわち、同図に示すように、通常使用される文字の解釈条件と共に、ユーザが希望する文字に対し、上述の解釈条件の設定が行われている。尚、本例では文字列解釈時の削除文字として上述の「★」、「☆」を設定し、数値解釈時の負符号文字として「●」を設定し、数値解釈時の小数点文字として「｜」を特別に設定した。

【００３９】次に、上述のように文字列解釈条件記憶エリア７ｂに設定した解釈条件に基づいて、レコード単位で入力する文字列の解釈処理を説明する。この処理は図２に示すように、先ずフォームオーバレイ印字を行うため、フォームオーバレイ印字を行おうとするフォームデータをＦＬ４ファイル１１ａ、ＥＧ４ファイル１１ｂから読み出す。例えば、この時前述の帳票フォームを使用したフォームオーバレイ印字を行う場合、図３の（ａ）に示したフォームデータを読み出す。またこの時、図３の（ｃ）に示すように、ＦＬＤファイル１３には帳票フォームの各項目に対応したフィールドが前述のように定義されており、プロセッサ２は帳票印刷プログラム７に従って定義されたフィールドに被フォームデータを書き込んでいく。

【００４０】以下、図８に示すフローチャートに従って処理される。先ず、アプリケーションソフト８によって作成されたＣＳＶデータが文字列データとして読み出される（ステップ（以下Ｓで示す）１）。次に、ＣＰＵは読み出されたデータが文字列データか、又は数値データか判断する（Ｓ２）。ここで、例えば読み出されたデータが文字列データであれば、不図示のカウンタ（ＣＮＴ）の計数値をＱＣＳ、すなわち前述の文字列解釈条件記憶エリア７ｂに記憶した文字列解釈時の削除文字数（ＱＣＳ）に設定する（Ｓ３）。本例の場合、図７に示すように４個の文字（「□」、「II」（半角スペース）、「★」、「☆」）が削除文字として登録されているため、上記処理（Ｓ３）において、「４」が設定される。

【００４１】この処理は上記４個の登録文字に対して個々に行われ、先ず文字列解釈条件設定エリア７ｂ（記憶エリア２５ａの＃１）から「□」の文字が読み出され（Ｓ４がＮ、Ｓ５）、削除文字の処理が行われる（Ｓ６）。ここで、先ず最初の削除処理では、例えば「□東□京」の文字列（ＣＳＶデータ）に対しては「□」の文字が削除され、「東京」と変換される。

【００４２】このようにして、読み出した文字列に対する「□」の文字の削除処理が完了すると、ＣＰＵはカウンタ（ＣＮＴ）のカウントダウンを行い（Ｓ７）、カウンタ（ＣＮＴ）の計数値が「０」でないか判断する（Ｓ４）。この時、カウンタ（ＣＮＴ）の値は「３」であるので、記憶エリア２５ａの次のエリア＃２から「II」（半角スペース）の文字が読み出され（Ｓ４がＮ、Ｓ５）、読み出された文字列に「II」（半角スペース）が含まれていれば上述と同様の処理を行う。

【００４３】次に、更にカウンタ（ＣＮＴ）をカウントダウンし、カウンタ（ＣＮＴ）のカウント値が「２」になると、次の削除文字として「★」を記憶エリア＃３から読み出す。この「★」はユーザの要求に応じて文字列解釈条件の記憶エリアに登録されたものであり、通常の解釈条件によっては削除処理できない文字である。例えば、図９に示す「★東★京」の文字列がＣＳＶデータとして供給された場合、従来では「★」を削除することができなかった。しかし、本例によれば文字列解釈時の削除文字として「★」が登録されているので、削除処理を行うことができ、「東京」と変換できる。

【００４４】以上の処理がすべて終了すると、例えば「★東□京」や「□東☆京」などの文字列も全て「東京」に変換される。さらに、「☆」の文字に対しても同様であり、ＣＳＶデータとして読み出した文字列に「☆」が存在する場合、例えば「☆東☆京」の文字列は「東京」に変換される。

【００４５】上述の処理が終了すると、カウンタ（ＣＮＴ）のカウント値は「０」となり（Ｓ４がＹ（イエス））、文字列データの出力処理を行う（Ｓ８）。すなわち、フィールド定義された文字列解釈時のフィールド３Ａ、４Ａ、５Ａ、６ＡにＣＳＶデータを書き込む。例えば、図３の（ｄ）の例によれば、フィールド３Ａに「札幌」の文字、フィールド４Ａに「東京」の文字、フィールド５Ａに「大阪」の文字、フィールド６Ａに「福岡」の文字がそれぞれ印字される。

【００４６】次に、読み出されたＣＳＶデータが数値データの場合について説明する。この場合、前述の判断（Ｓ２が数値）であり、処理（Ｓ９）に移行する。この処理（Ｓ９）は図７の数値解釈時の削除文字に含まれる文字数をカウンタ（ＣＮＴ）に設定するものであり、この場合「□」、「ＩＩ」（半角スペース）、「，」、「，」の４個である。したがって、この場合にもカウンタ（ＣＮＴ）には「４」が設定され（Ｓ９）、カウンタ（ＣＮＴ）のデータが「０」であるか判断（Ｓ１０）した後、削除文字「□」を読み出し（Ｓ１２）、前述と同じ削除処理を行う（Ｓ１３）。以下、同様にして個々の削除文字「ＩＩ」（半角スペース）、「，」、「，」に対する削除処理を行った後、次の負符号文字の処理に移行する（Ｓ１３、Ｓ１０～Ｓ１３、Ｓ１０がＹ）。

【００４７】次の負符号文字の変換処理においても、「－」（全角マイナス）、「-」（半角マイナス）、「◆」、「●」の４個の負符号文字が登録されているので、カウンタ（ＣＮＴ）値として「４」を設定し（Ｓ１４）、カウンタ（ＣＮＴ）のデータが「０」を判断（Ｓ１５）した後、負符号文字「－」を読み出し（Ｓ１６）、処理を行う（Ｓ１７）。この場合にも、ユーザが設定した特定の文字「●」が存在し、この文字を含む数値列に対し、負符号変換処理を行う（Ｓ１７）。例えば、図９に示す“●100,000 ”の数値列は「－100000」に変換される。

【００４８】さらに、小数点文字に対しても同様であり、カウンタ（ＣＮＴ）値として「３」を設定し（Ｓ１９）、順次、小数点文字「．」（全角ピリオド）、「.」（半角ピリオド）、「｜」に対する変換処理を行い（Ｓ２０～Ｓ２３）、例えば「0｜123 」の文字列データは「 0. 123 」に変換する。尚、以上の数値変換処理の後、２バイトの数値コードは実際の数字データに変換され（Ｓ２４）、数値データとして出力される（Ｓ２５、Ｓ２６）。

【００４９】尚、図３の（ｄ）は上述の帳票印刷プログラム処理の後、プリンタドライバ８を介して出力される印刷データによってプリンタ装置１５で印刷処理が行われ、その結果作成された帳票を示す。

【００５０】以上のように本例によれば、従来解釈できなかった「★」、「☆」や、「●」、「｜」に対しても、ユーザが希望する通りの変換処理を行うことができる。次に、上述の処理によりユーザの希望通りの変換処理によって得られた変換後のデータは正確なものである。したがって、帳票フォームに印字される被フォームデータ（ＣＳＶデータ）により色分け処理を行う場合、正確な色塗り処理を行うことができる。

【００５１】例えば、図１０に示す比較条件に従って帳票フォームのセルに色塗り加工を行う場合を考える。この場合、比較条件１に「＜」を設定し、被比較値（Ｓ）を「－１００」とする。同じく、比較条件２に「≧」を設定し、被比較値（Ｔ）を「－２００」とする。そして、ＣＳＶデータがこの間の値であれば緑色で対応するセルの背景を塗り分ける印字条件を設定する。尚、図１０に示す表示は前述と同様ディスプレイ５に行われ、例えばキーボード６を操作して設定する。尚、比較条件１、２等は、勿論他の数値であっても良い。

【００５２】上述の設定において、ＣＳＶデータとして“●120 ”の数値列や“●200 ”の数値が入力するものとすると、本例のシステムでは文字列解釈条件７ｂにおいて図７に示す解釈条件が設定されているので、直ちに「－120 」、「－200 」と判断できる。したがって、この場合上記ＣＳＶデータ（“●120 ”、及び“●200 ”）を印字するセル内は正確に緑色で塗られる。

【００５３】また、上述の例はＣＳＶデータが数値である場合を説明したが、地名であってもよく、例えば「東京」の地名を認識した時緑色で塗り分ける印字条件であれば、「★東★京」のＣＳＶデータに対し正確に「東京」を認識し、対応するセルを緑色に塗ることができる。

【００５４】また、数値や地名に限らず、商品名等のデータでもよく、又印刷条件も色分け処理に限るものではない。尚、本例では文字列解釈条件設定プログラム７ａにより削除文字や負符号文字、小数点文字に対して解釈条件の設定処理を行ったが、本例はこれらの解釈条件以外に対しても設定可能であることは勿論である。

【００５５】また、上述の例では本発明の装置、及びシステムをフォームオーバレイ印字に適用したが、フォームオーバレイ印字に限るものではなく、例えばプログラムが異なる他のフォームデータに対する印刷や、他の表計算ソフトやデータベースソフトへの適用も可能である。

【００５６】

【発明の効果】以上説明したように本発明によれば、ユーザの必要に応じて、通常の文字列解釈条件と異なる条件で特定の文字の解釈を行うことができ、ユーザにとって極めて利用性の高い記録装置、及び記録システムを提供できる。

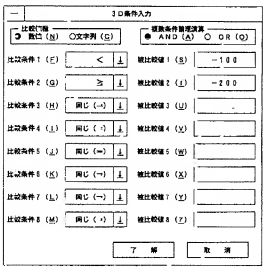
【００５７】また、ＣＳＶデータに対し、特別な文字を区切り文字として設定することにより、ＣＳＶデータに対しても桁区切りを行うことができ、ＣＳＶデータの利用性を増大することができる。

【００５８】さらに、本発明によれば、帳票印刷データのセルに予め定義された印刷加工条件とそのセル内に合成して書き込まれるＣＳＶデータとを比較し、その比較結果に応じて色分け加工処理を行うような合成印刷処理を行う場合、印刷加工条件に対して正確に比較判別が行われるようにＣＳＶデータの解釈条件を予め揃えることができ、高精度に印刷加工処理を行わせることが可能となる。

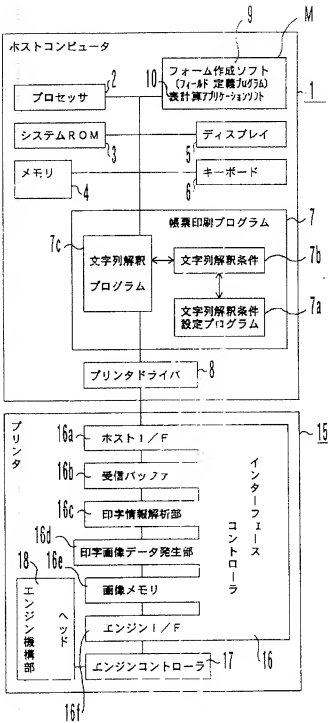
【図面の簡単な説明】
【図１】本実施形態例のシステムに説明する構成図である。
【図２】本実施形態例の機能を説明する図である。
【図３】本実施形態例の具体的な例として、帳票フォームの例を説明する図であり、（ａ）は帳票フォームを示し、（ｂ）はフィールド定義を示し、（ｃ）はフィールドと被フォームデータとの関係を示し、（ｄ）は帳票フォームへの被フォームデータの印刷結果を示す図である。
【図４】表計算アプリケーションソフト、及び帳票印刷プログラムの詳しい構成を説明するシステム構成図である。
【図５】文字列解釈条件記憶エリアのメモリマップを説明する図である。
【図６】文字列解釈条件記憶エリアに登録される文字例を示す。
【図７】入力文字列を文字列解釈条件に従って変換処理する具体例を説明する図である。
【図８】本例の処理を説明するフローチャートである。
【図９】本例の「文字列解釈条件」の例を示す図である。
【図１０】ＣＳＶデータに色分け等の印字条件を設定する場合の表示例を示す図である。
【図１１】従来の「文字列解釈条件」の一例を示す図である。
【図１２】従来の文字列３例を説明する図である。
【符号の説明】
１　パーソナルコンピュータ
２　プロセッサ
３　システムＲＯＭ
４　メモリ
５　ディスプレイ
６　キーボード
７　帳票印刷プログラム
７ａ　文字列解釈条件設定プログラム
７ｂ　文字列解釈条件
７ｃ　文字列解釈プログラム
８　プリンタドライバ
９　フォーム作成ソフト
１０　表計算アプリケーションソフト
１１ａ　ＦＬ４ファイル
１１ｂ　ＥＧ４ファイル
１２　フィールド定義プログラム
１３　ＦＬＤファイル
１５　プリンタ装置
１６ａ　ホストＩ／Ｆ
１６ｂ　受信バッファ
１６ｃ　印字情報解析部
１６ｄ　印字画像データ発生部
１６ｅ　画像メモリ
１６ｆ　エンジンＩ／Ｆ
１７　エンジンコントローラ
１８　エンジン機構部
２５、２５ａ～２５ｄ　記憶エリア
２６ａ　了解ボタン
２６ｂ　取消ボタン

【図６】

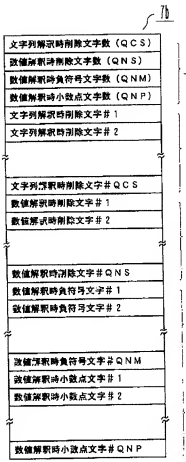
【図１０】



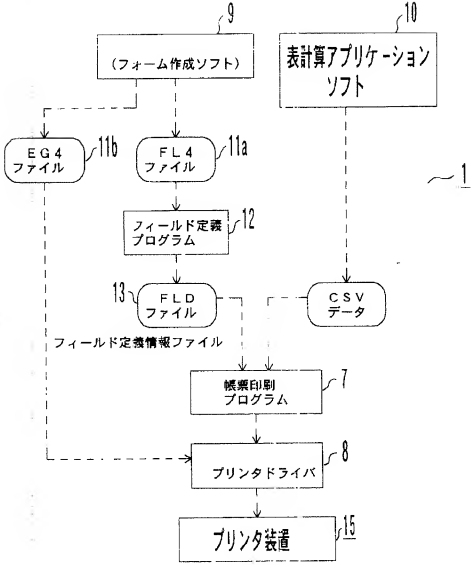
【図1】


9
M
ホストコンピュータ
フォーム作成ソフト
(フィールド 定義プログラム)
表計算アプリケーションソフト
10
1
プロセッサ
2
システムROM
3
メモリ
4
ディスプレイ
5
キーボード
6
帳票印刷プログラム
7
7c
文字列解釈
プログラム
文字列解釈条件
7b
文字列解釈条件
設定プログラム
7a
プリンタドライバ
8
プリンタ
15
16a
ホストI/F
16b
受信バッファ
16c
印字情報解析部
16d
印字画像データ発生部
16e
画像メモリ
エンジンI/F
インターフェース
コントローラ
16
18
エンジン機構部
ヘッド
エンジンコントローラ
17
16f

【図5】


7b
文字列解釈時削除文字数（QCS）
数値解釈時削除文字数（QNS）
数値解釈時負符号文字数（QNM）
数値解釈時小数点文字数（QNP）
25
文字列解釈時削除文字＃1
文字列解釈時削除文字＃2
25a
文字列解釈時削除文字＃QCS
数値解釈時削除文字＃1
数値解釈時削除文字＃2
25b
数値解釈時削除文字＃QNS
数値解釈時負符号文字＃1
数値解釈時負符号文字＃2
25c
数値解釈時負符号文字＃QNM
数値解釈時小数点文字＃1
数値解釈時小数点文字＃2
25d
数値解釈時小数点文字＃QNP

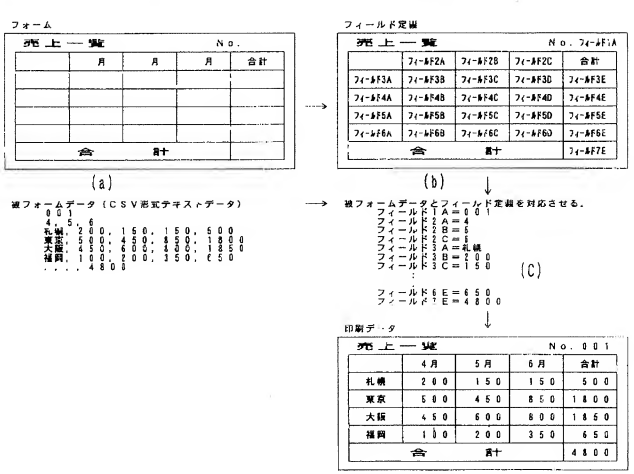
【図2】



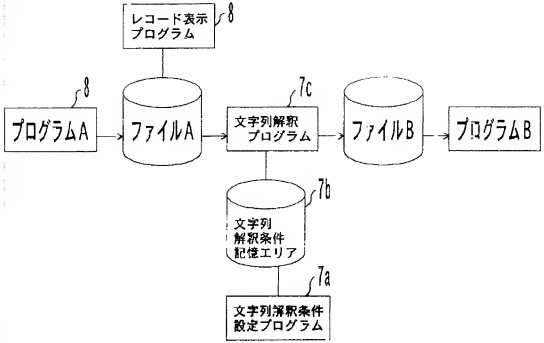
【図7】

文字列解釈時削除文字	□（全角スペース）、ll（半角スペース）、★、☆
数値解釈時削除文字	□（全角スペース）、ll（半角スペース）、，（全角カンマ）、，（半角カンマ）
数値解釈時負符号文字	－（全角マイナス）、-（半角マイナス）、◆（全角黒菱形）、●（全角黒丸）
数値解釈時小数点文字	．（全角ピリオド）、．（半角ピリオド）、｜（半角バー）

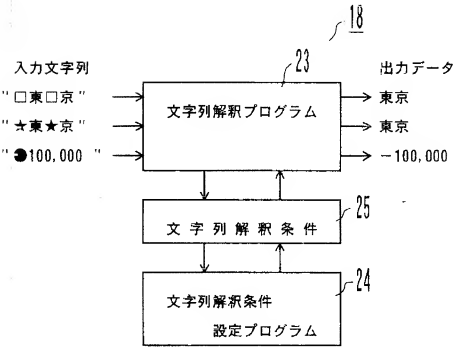
【図3】


フォーム
売上一覧
No.
月
月
月
合計
合計
(a)
フィールド定義
売上一覧
No. フィールドF1A
フィールドF2A
フィールドF2B
フィールドF2C
合計
フィールドF3A
フィールドF3B
フィールドF3C
フィールドF3D
フィールドF3E
フィールドF4A
フィールドF4B
フィールドF4C
フィールドF4D
フィールドF4E
フィールドF5A
フィールドF5B
フィールドF5C
フィールドF5D
フィールドF5E
フィールドF6A
フィールドF6B
フィールドF6C
フィールドF6D
フィールドF6E
合計
フィールドF7E
(b)
被フォームデータ（ＣＳＶ形式テキストデータ）
００１
４，５，６
札幌，２００，１５０，１５０，５００
東京，５００，４５０，８５０，１８００
大阪，４５０，６００，８００，１８５０
福岡，１００，２００，３５０，６５０
．．．，４８００
被フォームデータとフィールド定義を対応させる。
フィールドＩＡ＝００１
フィールドＦ２Ａ＝４
フィールドＦ２Ｂ＝５
フィールドＦ２Ｃ＝６
フィールドＦ３Ａ＝札幌
フィールドＦ３Ｂ＝２００
フィールドＦ３Ｃ＝１５０
：
フィールドＦ６Ｅ＝６５０
フィールドＦ７Ｅ＝４８００
(C)
印刷データ
売上一覧
No. ００１
４月
５月
６月
合計
札幌
２００
１５０
１５０
５００
東京
５００
４５０
８５０
１８００
大阪
４５０
６００
８００
１８５０
福岡
１００
２００
３５０
６５０
合計
４８００
(d)

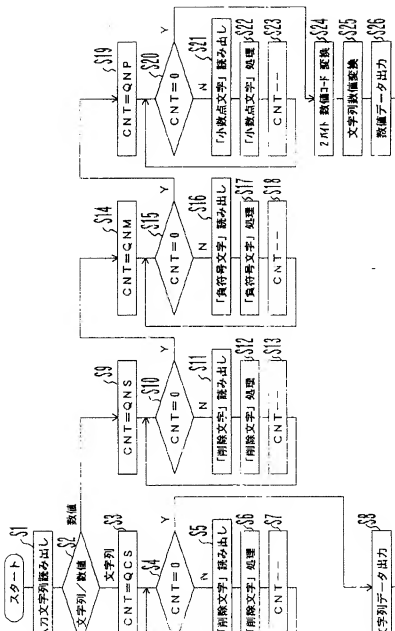
【図4】


レコード表示
プログラム
8
8
プログラムA
ファイルA
7c
文字列解釈
プログラム
ファイルB
プログラムB
7b
文字列
解釈条件
記憶エリア
7a
文字列解釈条件
設定プログラム

【図9】


18
入力文字列
23
出力データ
"□東□京"
"★東★京"
"●100,000"
文字列解釈プログラム
東京
東京
−100,000
文字列解釈条件
25
文字列解釈条件
設定プログラム
24

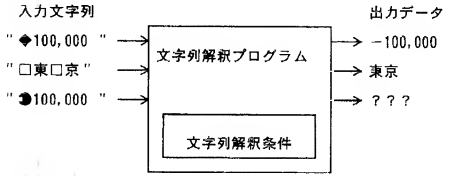
【図8】


スタート
S1
入力文字列読み出し
S2
文字列／数値
数値
文字列
S3
CNT=QCS
S4
CNT=0
Y
N
S5
「前段文字」読み出し
S6
「前段文字」処理
S7
CNT--
S9
CNT=QNS
S10
CNT=0
Y
N
S11
「前段文字」読み出し
S12
「前段文字」処理
S13
CNT--
S14
CNT=QNM
S15
CNT=0
Y
N
S16
「負符号文字」読み出し
S17
「負符号文字」処理
S18
CNT--
S19
CNT=QNP
S20
CNT=0
Y
N
S21
「小数点文字」読み出し
S22
「小数点文字」処理
S23
CNT--
S24
2バイト数値コード変換
S25
文字列数値変換
S26
数値データ出力
S8
文字列データ出力
エンド

【図11】

文字列解釈時削除文字	□（全角スペース）、II（半角スペース）
数値解釈時削除文字	□（全角スペース）、II（半角スペース）、，（全角カンマ）、，（半角カンマ）
数値解釈時負符号文字	－（全角マイナス）、-（半角マイナス）、◆（全角黒菱形）、△（全角白三角）
数値解釈時小数点文字	．（全角ピリオド）、.（半角ピリオド）

【図12】



(19) Japan Patent Office (JP)　(12) Japanese Unexamined Patent Application Publication (A)　(11) Japanese Unexamined Patent Application Publication Number H10-307817

(43) Publication date: November 17, 1998

(51) Int. Cl 5	Identification symbols	F1	G06F 15/20	566M
G06F 17/21		B41J 21/00		A
B41J 21/00		G06F 3/12		F
G06F 3/12			15/20	548E

Request for examination: Not yet requested　No. of claims: 4　OL　(Total of 13 pages)

(21) Application number　H9-118412

(22) Date of application　May 8, 1997

(71) Applicant: 000104124

Casio Electron MFG Co Ltd　4084 Miyadera, Iruma, Saitama, Japan

(71) Applicant: 000001443

Casio Computer Co Ltd　1-6-2 Honmachi, Shibuya-ku, Tokyo, Japan

(72) Inventor: Masaki Kobayashi

c/o Casio Electron MFG Co Ltd　2-229 Sakuragaoka, Higashiyamato, Tokyo

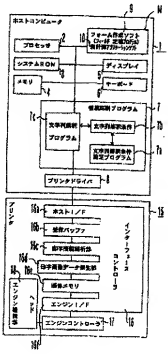
(74) Representative: Patent agent Yoshiyuki Osuga

(54) [Title of the Invention] Character String Conversion Device

(57) [Abstract]

[Objective] The present invention relates to a character string conversion device capable of setting interpreting conditions for a character string to be used at the time of executing form overlay printing using table calculation and database software, and is intended to provide a character string conversion device capable of accurately executing a printing work process such as a color coding process specifically based on data.

[Solving Means] In the present invention, at the time of creating a document by form overlay processing, the form data of the document is created, for example, by form creation software and CSV data created, for example, by a table calculation application is overlaid by a document printing program 7 as formed data. In this case, the character string of the CSV data is interpreted in accordance with the character interpreting conditions registered by a user, and form overlay printing is subsequently executed. Thus, the interpreting conditions required by the user may be set with respect to specific characters, and the process is executed in accordance with the setting, so color coding process or the like based on a numerical value or a character string may easily be executed.



[Scope of Patent Claims]

[Claim 1] A character string conversion device comprising:

a character string interpreting condition setting means for preliminarily setting interpreting conditions of a character string to be entered,

a character string interpreting means for interpreting character string data to be entered in accordance with interpreting conditions set by the character string interpreting condition setting means, and

a work processing means having an identifying means for identifying the character string entered to be printed in a specific printing region in accordance with predetermined specific conditions, for executing a specific printing work process according to the identified result of the identifying means, wherein

said identifying means identifies said entered character string based on the result interpreted by said character string interpreting means.

[Claim 2] A character string conversion device according to Claim 1, wherein said given conditions are defined in document definition data of form overlay printing.

[Claim 3] A character string conversion device according to Claim 1, wherein said character string consists of CSV data generated by table calculation application software.

[Claim 4] A character string conversion device, wherein data that has been subjected to the work process by said work processing means is output to a recording device or to a display device.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention] The present invention relates to a character string conversion device capable of setting interpreting conditions for characters to be used at the time of executing form overlay printing using table calculation or database software.

[0002]

[Prior Art] Conventionally, when running a computer using table calculation or database software, the processes are executed by interpreting a character string or a numerical value string including various symbols. Furthermore, for example, when executing form overlay printing by connecting a printer device to the computer, a printing process is executed by interpreting character strings, including various symbols.

[0003] In such a case, the process of interpreting a character string or numerical value string data (hereinafter, simply referred to as character string data) as described above, is processed by interpreting the character string data as an original meaning in accordance with a readily available program. Therefore, as for the process of interpreting the character string data, there is no choice but to follow the information of conditions required at the time of interpreting the character string that is readily available from the program.

[0004] For example, in the case of interpreting the character string "◆100,000", '◆' is normally interpreted as a minus sign, and the digit separation symbol ',' is ignored. Accordingly, in this case, the above character string "◆100,000" is interpreted as '–100000'. Moreover, in the character string "□To□kyo", '□' represents a double byte space, and the double byte space is normally ignored and, hence, interpreted as 'Tokyo', excluding the symbol '□'. Furthermore, Fig. 11 shows an example of 'character string interpreting conditions' that are normally executed.

[0005]

[Problem the Invention is Intended to Solve] As described above, in a conventional recording system,

in the case of executing an process of interpreting character string data, there has been no choice but an interpreting process in accordance with 'character string interpreting conditions' readily available in the program to be used. Therefore, expected results were not obtained when interpreting a character string that did not match the 'character string interpreting conditions' normally available in the program.

[0006] For example, in some character string data, if a symbol '●' is intended to be interpreted as a minus sign, that is, the character string "●100,000" is intended to be interpreted as '−100000', in a conventional example, the symbol of '●' is not defined in the 'character string interpreting conditions', so it is impossible to interpret "● 100,000" as '−100000'. Fig. 12 is a figure schematically showing the previous three examples. For character strings such as "◆100,000" or "□To□ kyo", an accurate interpreting process is possible, but for the character string "●100,000", an accurate interpretation is not possible.

[0007] On the other hand, the digit separation mark ',' is recognized as a separator of record units in CSV style format, for example. Consequently, representing a numerical value of long digits in such a format becomes a problem. Particularly, in the case of printing CSV data in a document form as formed data, sometimes a color coding process with respect to each item (each cell) or other printing work processes is executed according to the numerical value of the formed data (CSV data), so accurately reading the formed data is crucial.

[0008] For example, in the case of printing data under minus (−) 100,000 in each cell of a document form, if a work condition is already set to differentiate the background of the cell in red, it is necessary to recognize the minus (−) accurately. In such a case, as described above, if "◆100,000" or "●100,000" are not properly interpreted as '−100000', the appropriate color coding process may not be carried out.

[0009] Furthermore, not only the numerical values but also in the case of color coding work conditions with respect to place names included in CSV data, if it is not recognized as 'Tokyo', 'Sapporo', or 'Osaka', color coding work may not be carried out accurately.

[0010] Moreover, although the above example particularly describes document form and CSV data, the same thing applies in the case of other styles of forms and other styles of data. In order to meet the above objective, the present invention is intended to provide a character string conversion device capable of: optionally setting 'character string interpreting conditions' depending on the needs of each user; obtaining results expected by the user at the time of analyzing character string data; and accurately executing a printing work process such as color coding process and the like based on data.

[0011]

[Means for Solving the Problem] According to the invention described in Claim 1, the above problem is solved by providing a character string conversion device comprising a character string interpreting condition setting means for preliminarily setting interpreting conditions of the character string to be entered, a character string interpreting means for interpreting character string data to be entered in accordance with the interpreting conditions set by the character string interpreting condition setting means, and a work processing means having an identifying means for identifying the character string entered to be printed in a specific printing region in accordance with a predetermined specific conditions for executing specific printing work processes according to the identified result of the identifying means, wherein said identifying means identifies said entered character string based on the result interpreted by said character string interpreting means.

[0012] Herein, the character string interpreting condition setting means is, for example, to execute a setting process of interpreting conditions with respect to specific characters in accordance with a given program. Furthermore, unlike characters such as '□', '◆', or '、', the characters to be set here are to be set specifically according to the needs of the user. For example, '★' is set as a character to be deleted at the time of interpreting a character string, '●' is set as a minus sign character at the time of interpreting a numerical value, likewise '｜' is set as a decimal point character at the time of interpreting a numerical value.

[0013] With such a configuration, in a case that requires interpretation of a character string in accordance with interpreting conditions that are not the interpreting conditions normally set for the characters, that is, depending on the needs of the user, a specific character string may be interpreted under conditions that are different from normal character string interpreting conditions.

[0014] When the identifying means is to identify said entered character string based on the results interpreted by said character string interpreting means, for example, as in the example above, in the case of setting '●' to be interpreted as a minus sign character at the time of interpreting a numerical value, then '● 50' is interpreted as '−50'. Therefore, numerical values or character strings may be determined accurately, in the case of the above work processing. In other words, at the time of executing a color coding printing process according to the size of the numerical value or the content of the character string, the numerical value or the character string may be accurately determined, so the accurate color coding process or the like may be carried out.

[0015] The description in Claim 2 shows the invention described in the above Claim 1 more concretely, and in the configuration the given conditions are defined in the document definition data of form overlay printing.

[0016] In other words, in the event of form overlay printing, if the above character strings or the numerical values are included in the data to be supplied as formed data, with respect to the characters set by the user, corresponding interpretation is performed to execute the printing process, so for example, form overlay printing in accordance with printing conditions such as color coding processes according to the size of numerical values may be carried out.

[0017] The description in Claim 3 shows the invention described in the above Claim 1 more concretely, and said character string is CSV data generated by table calculation application software.

[0018] In other words, as for the CSV data, ',' is determined as a digit separation mark, but, for example, by the user preliminarily registering '｜' as a decimal point character at the time of interpreting a numerical value, the number of digits of the numerical value may be accurately recognized such as in the case of executing a color coding process according to the size of the numerical value as above. Thus, a color coding process may accurately be carried out even with respect to the CSV data.

[0019] The description of Claim 4 shows the invention described in the above Claim 1 more concretely. In the configuration, the output of the work processing is output to a recording device or to a display device. Therefore, for example, in the case of outputting to a recording device such as a printer device, an analyzing process is executed, such as in a printing information analyzing part within the recording device, and after having been rolled out once to the image memory, the printing output onto a recording paper will follow. Furthermore, in the case of outputting to a display device such as CRT display or the like, the display process is executed by

outputting, for example, to VRAM within the display device as display data.

[0020] Herein, for example, with respect to the display device as a case of displaying data subjected to a conversion process by the character string conversion device of the present invention, table calculation and the like may be displayed in addition to displaying form overlay images such as document forms and images in the database.

[0021]

[Embodiment of the Invention] Hereinafter, an embodiment of the present invention is described in detail using figures. Fig. 1 is a system configuration drawing describing a character string conversion device in the present embodiment. In the same figure, the present system is composed of a host computer 1 and a printer device 15. The host computer 1 is composed of a processor 2, a system ROM 3, a memory 4, a display 5, a keyboard 6, a document printing program 7, and a printer driver 8. Furthermore, the host computer 1 is equipped with a hard disc M, wherein utility software or application software such as form creation software (field definition program) 9, a table calculation application software 10, and/or the like are stored, and the processor 2 may process various information by executing these programs.

[0022] The processor 2 executes a process in accordance with the system programs stored in a system ROM 3. For example, the process of creating a document form is executed by reading out the form creation software 9 from a hard disc. Moreover, a table calculation process is executed by reading out the table calculation application software 10.

[0023] Furthermore, while the processor 2 is executing the above process, the data being generated is accommodated in a work area of the memory 4 to be used for an operation process or the like. When, required data is displayed on the display 5, and a key operation signal is supplied from the keyboard 6 to the processor 2.

[0024] Moreover, the document printing program 7 is composed of a character string interpreting condition setting program 7a, character string interpreting conditions 7b, and a character string interpreting program 7c. Furthermore, the concrete processes of the character string interpreting condition setting program 7a, the character string interpreting conditions 7b, and the character string interpreting program 7c are to be described later.

[0025] The printer driver 8 executes input/output control corresponding to the model of the printer device 15 to be connected to the host computer 1, and outputs the formed data to be described later to the printer device 15.

[0026] On the other hand, the printer device 15 is composed of an interface controller 16, an engine controller 17, and an engine mechanism part 18 including a printing head. The interface controller 16 is composed of a host interface (hereinafter, referred as host I/F) 16a, a receiving buffer 16b, a printing information analyzing part 16c, a printing image data generating part 16d, an image memory 16e, and an engine I/F 16f.

[0027] Fig. 2 is a figure particularly describing the functional configuration of the host computer 1. In other words, software or files shown in Fig. 2 represent a functional relation of the form creation software 9, the table calculation application software 10, and the document printing program 7 that are shown in Fig. 1. Furthermore, in the present embodiment, an example of form overlay printing is described, particularly as a formed data to be used for the form overlay printing. Data registered in the table calculation application 10 is used as CSV type data (hereinafter, referred as CSV data).

[0028] Accordingly, in the present example, the form creation software 9 is software for creating document

forms, or to be more specific, FL 4 file 11a and EG4 file 11b are created, as shown in Fig. 2. Herein, the FL4 file 11a is a form command file, and is a file for replacing the document form (figure shape of the document form) created by the form creation software 9 with a command from the form creation software, such as a description of text style. More specifically, at the time of creating a document form as shown in Fig. 3 (a), the line type of ruling lines or the coordinate positions of a rectangular shape or the like are the data, such as in the case of a rectangular shape, described as 'BOX 4.0, 2.0, 56.0, 26.0,...', and in the case of a ruling line, described as 'LINE 4.0, 4.0, 56.0, 4.0,...'.

[0029] On the other hand, the EG4 file 11b is a form data file, and the document form created by the form creation software 9 is stored by being replaced with the command of the printer device 15. More specifically, the description is in binary style data, such as in the case of a rectangular shape, described as '1Dh,00h,3Bh,00h,3Bh,40h,3Bh,...'.

[0030] Next, the field definition program 12 is a program for defining items to be written for formed data to be written in the above document form. More specifically, each item is defined as shown in Fig. 3 (b). For example, the region (field) of the number (No.) of a document form is field 1A. Furthermore, the region representing each month of the document form is field 2A through 2C—hereinafter, for each item respectively, field 3A to 3E, field 4A to 4E, field 5A to 5E, and field 6A to 6E, and so on. Moreover, field definition information files created by the above field definition program 12 are to be registered in a field file (FLD file) 13.

[0031] Fig. 4 is a figure describing the detailed functional configurations of the above table calculation application software 8 and the document printing program 7. Herein, the program A shows part of the processing program of the above table calculation application software 8, and the record display program also shows part of the configuration of the table calculation application software 8. Furthermore, file A shows a storing file of CSV data to be created by the table calculation application software 8. In other words, file A CSV data to be created by the table calculation application software 8 is facilitated, for example, by dividing data into record units and storing it in the form of divided data.

[0032] Moreover, the character string interpreting condition setting program 7a is a program for providing interpreting conditions of a character string, and is required at the time of interpreting character string data of record unit described above. Furthermore, the interpreting conditions to be set based on the character string interpreting condition setting program 7a is registered in the character string interpreting condition storing area 7b.

[0033] Fig. 5 is a figure for describing a memory map of the character string interpreting condition storing area 7b. As shown in the same figure, the character string interpreting condition storing area 7b is composed of each data storing area 25 of 'number of characters to be deleted at the time of interpreting a character string (QCS)', 'number of characters to be deleted at the time of interpreting a numerical value (QNS)', 'number of minus sign characters at the time of interpreting a numerical value (QNM)', and 'number of decimal point characters at the time of interpreting a numerical value (QNP)' and of storing area of specific characters of data to be stored in each of these storing areas. For example, the storing area 25a stores specific characters to be deleted at the time of interpreting a character string in #1, #2,...; the storing area 25b stores specific characters to be deleted at the time of interpreting a numerical value in #1, #2,...; the storing area 25c stores concrete minus sign characters at the time of interpreting a numerical value in #1, #2,..., and the storing area 25d

stores concrete decimal point characters at the time of interpreting a numerical value in #1, #2,....

[0034] The character string interpreting program 7c interprets a character string of CSV data to be stored in file A in accordance with the interpreting conditions stored in the above character string interpreting condition storing area 7b and writes the data after the interpretation in file B. Furthermore, the program B is a program for outputting the data written in file B to the printer driver 8 as shown in Figure 2.

[0035] In the system of the above configuration, first, a setting process of the character string interpreting conditions according to the character string interpreting conditions setting program 7a is described. This process is for executing the setting of characters requiring special interpreting conditions by operating the keyboard 6 provided with previously described host computer 1. More specifically, with the process of the character string interpreting condition setting program 7a, the display is presented on a display 4, such as the display shown in Fig. 6. In other words, in the present example, the setting process of four items: setting of characters to be deleted at the time of interpreting a character string, setting of characters to be deleted at the time of interpreting a numerical value, setting of minus sign characters at the time of interpreting a numerical value, and setting of decimal point characters at the time of interpreting a numerical value are executed with respect to each setting item.

[0036] For example, in the case of registering other characters as characters to be deleted other than '□' and 'II' (Note that 'II' indicates a single space. Hereinafter, a single space is represented in such a way for convenience.) that are normally used as characters to be deleted at the time of interpreting a character string, characters required to be registered are set, such as '★' and '☆', in a column of characters to be deleted in the character string (S) shown in Fig. 6. Furthermore, the same process is applied to characters for the deletion (N) of numerical values. If there are characters that a user requires to be registered as characters to be deleted, then the characters are to be set. However, in the description of the present example, it is presumed that there is no setting of characters to be deleted with respect to a special numerical value string desired by the user.

[0037] Next, a process is executed for setting a numerical value minus sign characters. Herein, for example, other than the normally used '◆' as a minus sign character, if '●' is intended to be set as a minus sign character, '●' is set for the desired registration in the column of numerical value minus sign character (M) shown in Fig. 6. Furthermore, in the case of setting ' | ' as a numerical value decimal point character, the process of setting the ' | ' is executed for the desired registration in the column of numerical value decimal point character (P) shown in the same figure.

[0038] Furthermore, in the event of the above setting process, if data is incorrectly entered, then the input operations up until then may be cancelled and return by pressing a 'cancel' button 26b shown in the same figure. When the interpreting condition setting process is finished as above, the process is ended by pressing the 'OK' button 26a. Accordingly, due to the process described above, in the character string interpreting condition storing area 7b, the interpreting condition shown in Fig. 7 is set. In other words, as described in the same figure, along with interpreting conditions of characters that are normally used, the above setting of the interpreting condition is established with respect to the characters desired by the user. Moreover, in the present example, the above '★' and '☆' were set as characters to be deleted at the time of interpreting a character string; '●' was set as a minus sign character at the time of

interpreting a numerical value; and '.' was specially set as a decimal point character at the time of interpreting a numerical value.

[0039] Next, as described above, based on the interpreting conditions set in the character string interpreting condition storing area 7b, the process of interpreting a character string to be entered by record unit is described. As shown in Fig. 2, in this process, first, since form overlay printing is executed, the form data in the subject of form overlay printing is read out from the FL4 file 11a and EG4 file 11b. For example, then, in the case of executing form overlay printing using the document form that has previously been described, the form data shown in Fig. 3 (a) is read out. Furthermore, at this time, as shown in Fig. 3 (c), in the FLD file 13, a field corresponding to each item of the document form is defined as previously described, according to the document printing program 7 the processor 2 starts writing formed data in the defined field.

[0040] Hereinafter, the processes are executed in accordance with the flow chart shown in Fig. 8. First, the CSV data created by application software 8 is read out as character string data (Step (hereinafter, indicated as S) 1). Next, the CPU determines whether the data read out is character string data or numerical value data (S2). Herein, for example, if the data read out was character string data, the count of the counter (CNT) (not illustrated) is set as QCS—that is, set as the number of characters to be deleted (QCS) at the time of interpreting a character string stored in the previously described character string interpreting condition storing area 7b (S3). In this case, as shown in Fig. 7, since four letters ('□','ll' (single byte space), '★', and '☆') have been registered as characters to be deleted, '4' is set in the above process (S3).

[0041] This process is executed individually with respect to the four of the above registered characters, first, from the character string interpreting condition setting area 7b (#1 of the storing area 25a) a character of '□' is read out (S4 is N, S5), and the process of character deletion is executed (S6). Herein, in the very first deleting process, for example, with respect to a character string (CSV data) of '□To□kyo', the character of '□' is deleted and converted to 'Tokyo'.

[0042] In this way, once the process of deleting the character '□' with respect to the character string that has been read out is complete, the CPU executes the countdown of the counter (CNT) (S7) to determine whether the count of the counter (CNT) is '0' (S4). Then, the value of the counter (CNT) is '3', so from the subsequent area #2 of the storing area 25a, the character 'll' (single byte space) is read out (S4 is N, S5), and if the 'll' (single byte space) is included in the character string that has been read out, the same process is executed as above.

[0043] Next, the counter (CNT) is subjected to further countdown, and when the count value of the counter (CNT) becomes '2', then as the next character to be deleted, '★' is read out from the storing area #3. The '★' has been registered in the storing area of character string interpreting conditions in response to the demand of the user, so it is a character that may not undergo the deleting process under normal interpreting conditions. For example, if the character string '★To★kyo' shown in Fig. 9 is supplied as CSV data, conventionally, the '★' cannot be deleted. However, according to the present example, the '★' has been registered as a character to be deleted at the time of interpreting a character string, and the deleting process thereby becomes possible, so the string is converted to 'Tokyo'.

[0044] After all the above processes have finished, character strings such as '★To□kyo' or '□To☆kyo' for example, are all converted to 'Tokyo'. Moreover, the same is applied to the character '☆', in the case in which '☆' is present in a character string read out

as CSV data, such as the character string '☆To☆kyo' which is convertible to 'Tokyo'.

[0045] Once the above process is finished, the counted value of the counter (CNT) becomes '0' (S4 is Y(Yes)), and the output process of the character string data is executed (S8). In other words, the CSV data is written in the fields 3A, 4A, 5A, and 6A whose fields were defined at the time of interpreting the character string. For example, according to the example in Fig. 3 (d) in the field 3A characters of 'Sapporo', the field 4A characters of 'Tokyo', the field 5A characters of 'Osaka', and the field 6A characters of 'Fukuoka' are to be respectively printed.

[0046] Next, a case in which the CSV data that has been read out is numerical value data is described. In this case, as determined as previously described (S2 is a numerical value), move to the process (S9). This process (S9) is the setting of the number of characters included in the characters to be deleted at the time of interpreting the numerical value in Fig. 7 at the counter (CNT)—in this case, four of '□', 'll' (single byte space), '　,' and ', '. Therefore, also in this case, '4' is set in the counter (CNT) (S9), and after it is determined whether data in the counter (CNT) is '0' (S10), the character(s) to be deleted '□' is read out (S12), and the same deleting process as described previously is executed (S13). Hereinafter, likewise, after executing the deleting process with respect to the individual character 'll' (single byte space), '　,' and ', ', move to the following process of a minus sign character (S13, S10 through S13, S10 is Y).

[0047] Also, in the conversion process of the following four minus sign characters, since '—' (double byte minus), '–' (single byte minus), '◆', and '●' have been registered, the counter (CNT) value is set to '4' (S14), and the data on the counter (CNT) determines '0' (S15), and the minus sign character '–' is read out (S16) for the process to be executed (S17). Furthermore, in this case, the specific character '●' set by the user is present, and the minus sign conversion process is executed with respect to the numerical value string including the character (S17). For example, the numerical value string "●100,000" shown in Fig. 9 is converted to '–100000'.

[0048] Furthermore, the same applies to decimal point characters, as the counter (CNT) value is set to '3' (S19), and then successively, a conversion process with respect to the decimal point characters '．' (double byte period), '.' (single byte period), and '｜' are executed (S20 through S23), the character string data '0｜123' for example, is converted to '0.123'. Moreover, after the above numerical value conversion process, a numeral code of 2 bytes is converted to the actual numeral data (S24) and is output as the numerical value data (S25, S26).

[0049] Furthermore, Fig. 3 (d) shows a document created as a result of a printing process by the printer device 15 based on the printing data to be output through the printer driver 8 after the above document printing program process.

[0050] According to the present example as described above, even for '★', '☆', or '●' and '｜' which previously could not be interpreted, the conversion process may be carried out as desired by the user. Next, the post conversion data obtained by the conversion process as desired by the user according to the above process is accurate. Therefore, an accurate color coding process may be performed in the event of executing a color coding process with formed data (CSV data) to be printed on a document form.

[0051] For example, consideration is given to a case in which coloring work is executed for the cells in a document form in accordance with the comparison conditions shown in Fig. 10. In this case, in the comparison condition 1 '<' is set and the compared value (S) is prescribed as '–100'. Likewise, in the

comparison condition 2 '≧' is set and the compared value (T) is prescribed as '–200'. Then, if the CSV data falls in the values in between, a printing condition to differentiate the background of the corresponding cell in green is set. Moreover, the display to be shown in Fig. 10 takes place on the display 5 as previously described. For example, the keyboard 6 is operated for the setting. Furthermore, needless to say, the comparison conditions 1, 2, and so on, may be other numerical values.

[0052] In the above setting, if a numerical value string of "●120" or a numerical value of "●200" are to be entered as CSV data, since the interpreting conditions shown in Fig. 7 have been set in the character string interpreting conditions 7b with the system in the present example, the determination is made immediately as '–120' and '–200'. Therefore, in this case inside the cell where the above CSV data ("●120" and "●200") are to be printed would be accurately colored in green.

[0053] Furthermore, the above example describes a case in which the CSV data is a numerical value, but this may be a place name. For example, if the printing condition to differentiate with green color when the place name 'Tokyo' is recognized, with respect to the CSV data of '★To★kyo', 'Tokyo' is accurately recognized, and the corresponding cell may be colored in green.

[0054] Furthermore, instead of being limited to numerical values or place names, this may also be applied to data such as product names or the like, and the printing condition is similarly not limited to a color coding process. Moreover, in the present example, the process of setting interpreting conditions is executed in accordance with the character string interpreting condition setting program 7a with respect to characters to be deleted, minus sign characters, and to decimal point characters, but needless to say, the present example may be set for other interpreting conditions.

[0055] Moreover, in the above example, the device and the system of the present invention is applied to form overlay printing, but without being limited to form overlay printing, for example, printing of other form data with different programs or application to other table calculation software or database software is possible.

[0056]

[Effects of the Invention] As described thus far, according to the present invention, specific characters may be interpreted under the conditions different from normal character string interpreting conditions in response to the needs of a user, so it is possible to provide a recording device and recording system that are highly available to the user.

[0057] Furthermore, with respect to CSV data, by setting special characters as separation characters, digit separation may be executed even on the CSV data, thereby increasing availability of the CSV data.

[0058] Moreover, according to the present invention, by comparing the printing work condition primarily defined for a cell of document printing data and CSV data to be written in the cell by overlaying, depending on the result of the comparison, when an overlaying printing process is to be executed such as color coding work process, the interpreting conditions of CSV data may be arranged in advance so that the comparison determination is carried out accurately with respect to the printing work condition, thereby making it possible to execute a highly accurate printing work process.

[Brief Description of the Drawings]

[Figure 1] A configuration drawing describing the system of the present embodiment

[Figure 2] A figure describing functions of the present embodiment

[Figure 3] A figure describing an example of a document form, as a concrete example of the present embodiment: (a) showing a document form, (b) showing a field definition, (c) showing the relationship between the field and formed data, and (d) showing the printed result of the formed data for a document form.

[Figure 4] A system configuration drawing describing the configuration of table calculation software and document printing program in detail

[Figure 5] A figure describing the memory map of a character string interpreting condition storing area

[Figure 6] Character strings to be registered in the character string interpreting condition storing area are shown.

[Figure 7] A figure describing a concrete example of entered character strings to be subjected to a conversion process in accordance with the character string interpreting conditions.

[Figure 8] A flow chart describing a process of the present invention

[Figure 9] A drawing illustrating an example of 'character string interpreting conditions' in the present example

[Figure 10] A figure showing a display example in the event of setting printing conditions for CSV data such as color coding and the like

[Figure 11] A figure showing one example of conventional 'character string interpreting conditions'

[Figure 12] A drawing describing three examples of conventional character strings

[Explanation of the Symbols]

1	personal computer
2	processor
3	system ROM
4	memory
5	display
6	keyboard
7	document printing program
7a	character string interpreting condition setting program
7b	character string interpreting condition
7c	character string interpreting program
8	printer driver
9	form creation software
10	table calculation application software
11a	FL4 file
11b	EG4 file
12	field definition program
13	FLD file
15	printer device
16a	host I/F
16b	receiving buffer
16c	printing information analyzing part
16d	printing image data generating part
16e	image memory
16f	engine I/F
17	engine controller
18	engine mechanism part
25, 25a through 25d	storing area
26a	OK button
26b	cancel button

FIG. 1

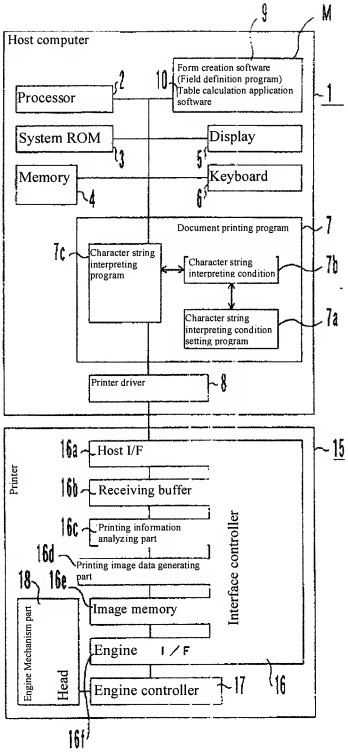

9
M
Host computer
Form creation software
(Field definition program)
Table calculation application
software
10
Processor
2
1
System ROM
3
Display
5
Memory
4
Keyboard
6
Document printing program
7
7c
Character string
interpreting
program
Character string
interpretine condition
7b
Character string
interpreting condition
setting program
7a
Printer driver
8
Printer
16a
Host I/F
15
16b
Receiving buffer
16c
Printing information
analyzing part
16d
Printing image data generating
part
18
16e
Image memory
Interface controller
Engine
I / F
Engine Mechanism part
Head
Engine controller
17
16
16f

FIG. 2

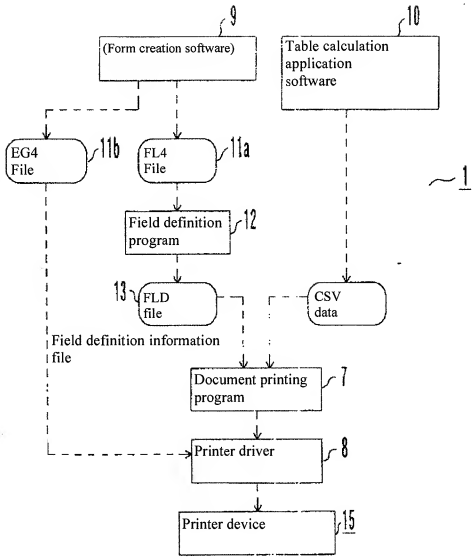


FIG. 7

Characters to be deleted at the time of interpreting character string	□ (Double byte space) 、 11 (Single byte space) 、 ★、 ☆
Characters to be deleted at the time of interpreting numerical	(Double byte space) 、 11 (Single byte space) 、 , (Double byte comma) 、 , (Single byte comma)
Minus sign characters at the time of interpretine numerical value	(Double byte minus) 、 - (Single byte minus) 、 ◆ (Double byte black) 、 ● (Double byte black circle)
Decimal point characters at the time of interpreting numerical value	(Double byte period) 、 . (Single byte period) 、 | (Single byte bar)

FIG. 3

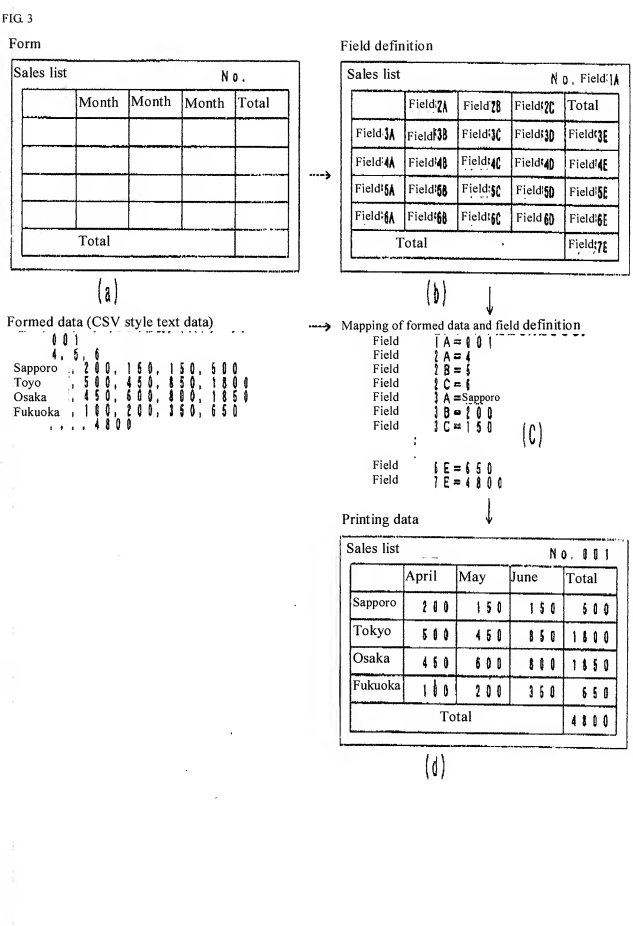

Form
Sales list
No.
Month
Month
Month
Total
Total
(a)
Field definition
Sales list
No. Field 1A
Field 2A
Field 2B
Field 2C
Total
Field 3A
Field 3B
Field 3C
Field 3D
Field 3E
Field 4A
Field 4B
Field 4C
Field 4D
Field 4E
Field 5A
Field 5B
Field 5C
Field 5D
Field 5E
Field 6A
Field 6B
Field 6C
Field 6D
Field 6E
Total
Field 7E
(b)
Formed data (CSV style text data)
001
4, 5, 6
Sapporo, 200, 150, 500
Toyo, 500, 450, 850, 1800
Osaka, 450, 600, 800, 1850
Fukuoka, 100, 200, 350, 650
, , , , 4800
Mapping of formed data and field definition
Field 1A = 001
Field 2A = 4
Field 2B = 5
Field 2C = 6
Field 3A = Sapporo
Field 3B = 200
Field 3C = 150
Field 6E = 650
Field 7E = 4800
(C)
Printing data
Sales list
No. 001
April
May
June
Total
Sapporo
200
150
150
500
Tokyo
500
450
850
1800
Osaka
450
600
800
1850
Fukuoka
100
200
350
650
Total
4800
(d)

FIG. 4

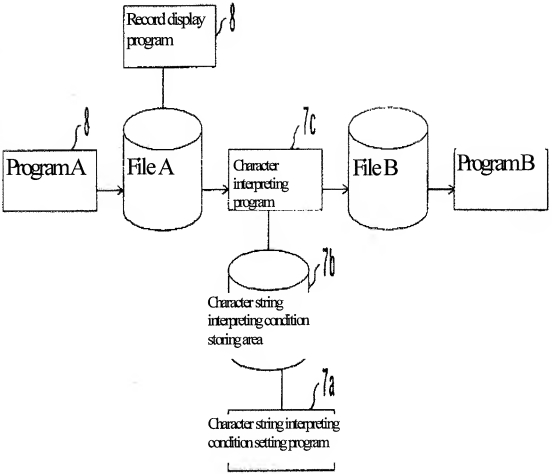

Record display program
8
8
Program A
File A
7c
Character interpreting program
File B
Program B
7b
Character string interpreting condition storing area
7a
Character string interpreting condition setting program

FIG. 5

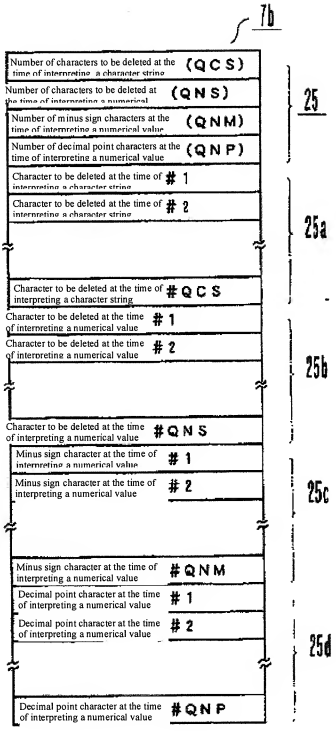

7b
Number of characters to be deleted at the time of interpreting a character string (QCS)
Number of characters to be deleted at the time of interpreting a numerical (QNS)
Number of minus sign characters at the time of interpreting a numerical value (QNM)
Number of decimal point characters at the time of interpreting a numerical value (QNP)
25
Character to be deleted at the time of interpreting a character string # 1
Character to be deleted at the time of interpreting a character string # 2
25a
Character to be deleted at the time of interpreting a character string #QCS
Character to be deleted at the time of interpreting a numerical value # 1
Character to be deleted at the time of interpreting a numerical value # 2
25b
Character to be deleted at the time of interpreting a numerical value #QNS
Minus sign character at the time of interpreting a numerical value # 1
Minus sign character at the time of interpreting a numerical value # 2
25c
Minus sign character at the time of interpreting a numerical value #QNM
Decimal point character at the time of interpreting a numerical value # 1
Decimal point character at the time of interpreting a numerical value # 2
25d
Decimal point character at the time of interpreting a numerical value #QNP

FIG. 6

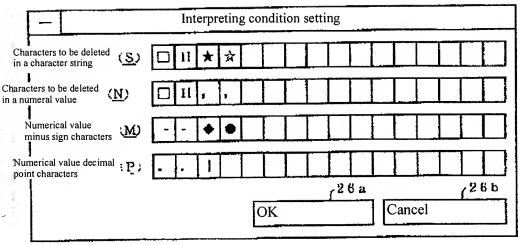

Interpreting condition setting
Characters to be deleted in a character string (S)
Characters to be deleted in a numeral value (N)
Numerical value minus sign characters (M)
Numerical value decimal point characters (P)
26a
26b
OK
Cancel

FIG. 8

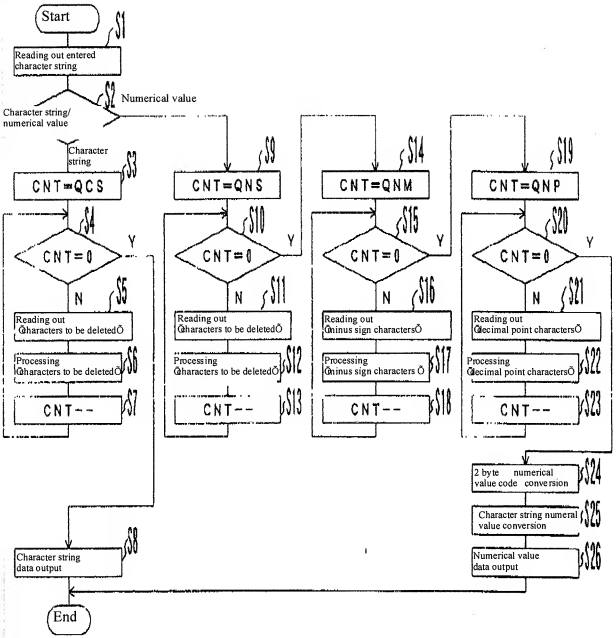

Start
S1
Reading out entered character string
S2
Numerical value
Character string/ numerical value
Character string
S3
CNT=QCS
S4
CNT=0
Y
N
S5
Reading out Characters to be deleted
S6
Processing Characters to be deleted
S7
CNT--
S9
CNT=QNS
S10
CNT=0
Y
N
S11
Reading out Characters to be deleted
S12
Processing Characters to be deleted
S13
CNT--
S14
CNT=QNM
S15
CNT=0
Y
N
S16
Reading out Minus sign characters
S17
Processing Minus sign characters
S18
CNT--
S19
CNT=QNP
S20
CNT=0
Y
N
S21
Reading out Decimal point characters
S22
Processing Decimal point characters
S23
CNT--
S24
2 byte numerical value code conversion
S25
Character string numeral value conversion
S26
Numerical value data output
S8
Character string data output
End

FIG. 9

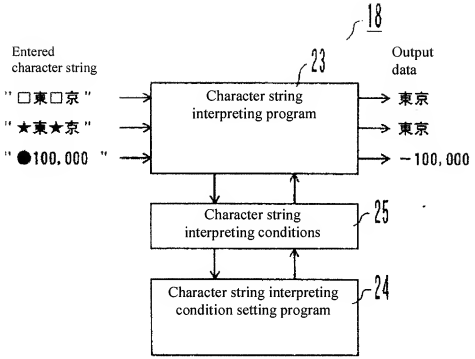

18
Entered character string
23
Output data
" □東□京 "
" ★東★京 "
" ●100,000 "
Character string interpreting program
東京
東京
−100,000
Character string interpreting conditions
25
Character string interpreting condition setting program
24

FIG. 10

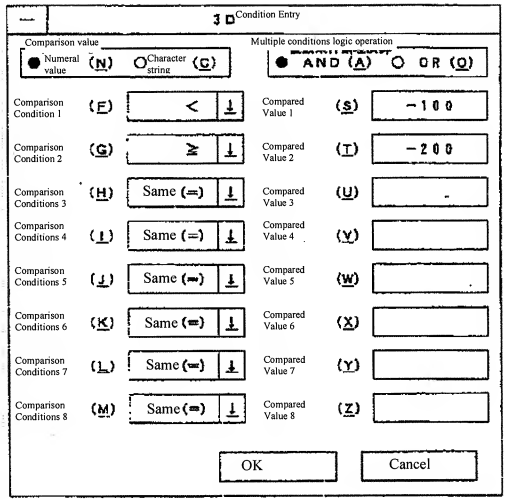

30 Condition Entry
Comparison value
Numeral value (N)
Character string (C)
Multiple conditions logic operation
AND (A)
OR (O)
Comparison Condition 1 (F) <
Compared Value 1 (S) -100
Comparison Condition 2 (G) ≧
Compared Value 2 (T) -200
Comparison Conditions 3 (H) Same (=)
Compared Value 3 (U)
Comparison Conditions 4 (I) Same (=)
Compared Value 4 (V)
Comparison Conditions 5 (J) Same (=)
Compared Value 5 (W)
Comparison Conditions 6 (K) Same (=)
Compared Value 6 (X)
Comparison Conditions 7 (L) Same (=)
Compared Value 7 (Y)
Comparison Conditions 8 (M) Same (=)
Compared Value 8 (Z)
OK
Cancel

FIG. 11

Characters to be deleted at the time of interpreting character string	Double byte space 、 Single byte space
Characters to be deleted at the time of interpreting numerical value	Double byte space 、 Single byte space 、 Double byte comma 、 Single byte comma
Minus sign characters at the time of interpreting numerical value	Double byte minus 、 Single byte minus 、 Double byte black diamond 、 Double byte white triangle
Decimal point characters at the time of interpreting numerical	Double byte period 、 Single byte period

FIG. 12